An Adaptation Algorithm for Adaptive Web Based Systems based on Link Structure and Document Similarity

BUJAR RAUF
South East European University
Dpt. of Computer Science
Ilindenska nn, 1200 Tetovo
MACEDONIA
b.raufi@seeu.edu.mk

JULIANA GEORGIEVA
Technical University of Sofia
Fac. of Computers and Control Systems
kl. Ohridski 8, 1000 Sofia
BULGARIA
july@tu-sofia.bg

ARTAN LUMA
South East European University
Dpt. of Computer Science
Ilindenska nn, 1200 Tetovo
MACEDONIA
a.luma@seeu.edu.mk

FLORJE ISMAILI
South East European University
Dpt. of Computer Science
Ilindenska nn, 1200 Tetovo
MACEDONIA
f.ismaili@seeu.edu.mk

XHEMAL ZENUNI
South East European University
Dpt. of Computer Science
Ilindenska nn, 1200 Tetovo
MACEDONIA
xh.zenuni@seeu.edu.mk

Abstract: Many traditional adaptive web based systems today are complex in its design and implementation which makes them hard to implement in general purpose web sites. The lack of efficient approach and adaptation algorithm makes their usage in isolated cases like: e-commerce sites or only education systems. In this paper we introduce a new adaptation algorithm based on web sites link structure and document similarities both represented through matrices. Initial results show good performance of the algorithm for both per-user based adaptation and adaptation for groups of users.

Key–Words: Adaptive web based systems, adaptation algorithm, document similarity, data mining

1 Introduction

Many of the traditional web pages used today suffer from a "one-size-fits-all" [1] problem. This means that when a user accesses a web page of a library, it will be presented with the same set of information every time it visits it. A tourist that tries to find information about travel and hotel arrangements will be presented with the same order of information regardless of its destination preferences [1]. The same issue can be generalized for many, but not all, web applications in various domains like: e-commerce, corporate web sites and virtual education. The solution for overcoming this situation is the development of user-adaptive software systems [1] or Adaptive Web Sites. Adaptive Web Sites represent Web-Based Systems that tend to arrange their content, structure or both based on user access preferences.

Starting as a pioneering work of a few research communities in the middle of 90’s [2] today it represents a research milestone for many communities involving various research disciplines like: user modeling, Web Usage Mining, natural language processing, intelligent tutoring systems, cognitive sciences and semantic web. The adaptation in such systems is done by building a model of goals, knowledge and preferences as well as using this for interaction with the user during its browsing. In traditional adaptive web based systems such an interaction is done by building a user model and applying the same for adapting the content for the purpose of the user itself [3]. This relationship between system, user and adaptation is presented by Brusilovsky in the form of user modeling-adaptation loop [3]. In this approach, the system collects data about user, processes the same and finally results are presented in an adapted way. The adaptation process, as stated by Brusilovsky, is performed through the adaptation methods and adaptation techniques[3].

By adaptation methods we consider a more generalized and abstracted approach concerning adaptation. This usually represents a description of what the adaptation should do rather than how to do it. In adaptation methods, the implementation per se can be done in various ways. Methods usually include a more abstracted representation of adaptation rules such as: ”insert a certain text fragment in a page if a certain concept is been visited by the user etc”. In adaptation techniques on the other hand, a more detailed and well specified approach for delivering adaptation is used.
and consequently, as such, are part of the implementation process. Each adaptation technique is characterized by specific knowledge representation, modeling, adaptation engine etc [3].

There are two distinctive levels on the sense where adaptation can be performed. The first one is known as content-level adaptation where the adaptation is performed on presentation of content and the second is recognized as link-level adaptation where adaptation focuses on link structure manipulation. The former is known as Adaptive Presentation and the latter is accepted as Adaptive Navigation Support [3].

The rest of the paper is structured as follows: in section 2 a brief introduction on related works done regarding the adaptation approaches and algorithms in adaptive web based systems is given. In section 3, the preliminary requirements for our adaptive algorithm as well as our proposed algorithm’s pseudocode is outlined. Section 4 introduces some experimental results and finally section 5 concludes on current results and outlines the future steps.

2 Related Work

Most of the adaptive web based systems developed are described as integral systems consisted of many modules such as user profilers, log and web usage miners, content managers, web site authoring tools and information acquisitioners and searchers [6]. However, in many approaches, there is no clear description of an algorithm that performs the adaptation process. De Bra for example, mentions a technique called conditional object inclusion [4] where different pieces of information are included in the page based on certain conditions.

Many of the developed tools utilize WebML as a language for defining the concept level of the domain used as well as some tools for automatic code generation for expressing the so called Event-Condition-Action aspects of their application [7]. The Event-Condition-Action (ECA) represent constructs for specifying rules unto which the events are triggered depending on some condition, resulting ultimately in a certain action. This approach has been used in Daniel et al [7], with the utilization of Chimera Exception language. However, this approach lacks an algorithmic idea, i.e. there is no true algorithm used and all the adaptation is done through ECA constructs of Chimera-Exception language.

Some other approaches for adaptation strive to be preference based constraint optimization techniques which are derived from qualitative decision making theories. One such approach is used by Brafman et al. [8] with the application of CP-Nets (Conditional Preference Networks). In this approach, the user preferences are constructed as CP-Nets instances in the form of most desirable presentation that satisfies certain user requests. The construction of such desirable presentations is done through conditional probabilities of two consequent pages. For example, if a user is asked to describe page A and page B then the preference for page C is set, so we have a conditional probability between pages. The problem with the approach is that the system does not reason in uncertainty.

The above mentioned approaches deal mostly with content adaptation or as mentioned in section I, adaptive content presentation. However, there are approaches that are more adaptive navigation support oriented [6], [9], [10]. Doerr et al., seeks the possibility of adaptive navigation support for simplifying web traversals by means of web log analysis and navigation pattern creation. This web traversal simplification is done through optimization of patterns for traversing the navigation graph through skipping irrelevant graph nodes, skipping irrelevant graph nodes on per-user basis, providing shortcuts to popular documents and reacting to temporal phenomena [10]. Another approach [9] extends the above mentioned by adding other data mining techniques used for further adaptation such as association rule mining, cluster mining or sequential pattern mining [6]. The main drawbacks of all the above mentioned proposals can be outlined as follows:

1. In all the approaches mentioned above there is no clear focus on adaptation formalism and algorithm on how the adaptation is performed?
2. Many of the data mining techniques used in overhead contributions focus extensively on user navigation patterns (the importance of which it should not be diminished) rather than what the documents itself represents and to whom they are related
3. Some of the approaches use definition of rules how and when the adaptation should occur which sometimes it represents a tedious authoring task. Like in [8] where the author decides about relevant information based subjective judgment from the web site administrator, rather than real user preferences. Our proposed approach, as is going to be presented in sections below, completely omits this step.

Answers to the above mentioned drawbacks are aimed in the sections to follow where a mathematical foundation, implementation of a new adaptation algorithm as well as experimental results are going to be presented.
3 A New Approach on Adaptation for Adaptive Web Based Systems

It has been mentioned earlier that designing and implementing an Adaptive Web Based system is not a trivial task and it involves many disciplines from knowledge management to artificial intelligence. For this reason, many frameworks have been designed to address the issue like AHA! [11], SKILL [12], Multi-book [13], ACE [14], ART-Web [15], MetaLinks [16], LAOS Framework for Authoring Adaptive Hypermedia Systems [17] and GOMAWE [18]. In [21] we have introduced a framework for adaptive web based systems using layered approach to address the complexities that arise from implementing such systems. The framework is consisted of five layers namely: data layer, concept layer, user layer, adaptation layer and presentation layer. The concept layer was elaborated on [21] while the user layer on [19] and this section is going to get focused on adaptation layer with the new algorithmic approach on adaptation. The overall adaptation process proposed in our approach is consisted of three main constituents as depicted in fig 1.

The indexer creates the vector space model [22] of each document that web system is consisted of as well as keeps the relationship between real documents and vector space model. For vector space generation we utilize term frequency - inverse document frequency (tf-idf) and Latent Semantic Indexing (LSI). The similarity matrix generates the numerical relationships between documents represented in a matrix form. For this purpose Jaccard and Cosine Similarity is used.

3.1 The adaptation algorithm

After the creation of vector space model and of document similarity matrix, the adaption algorithm rearranges the link structure and content for every particular user or groups of users. Our adaptation algorithm manipulates two matrices necessary for adaptation. The first one is the link matrix which represents an adjacency matrix of the entire link structure of the hypertext system and the second one, the similarity matrix, deals with similarity measures of its documents. These two matrices and their properties are formalized as below.

Let $L$ represent a matrix where the relationships between pages or atomic pieces of information [19], [20] are given in the form:

$$L = \begin{pmatrix} l_{1,1} & \cdots & l_{1,n} \\ \vdots & \ddots & \vdots \\ l_{n,1} & \cdots & l_{n,n} \end{pmatrix}$$

From this matrix, by using similarity measures between documents a new matrix can be generated represented as:

$$S = \begin{pmatrix} s_{1,1} & \cdots & s_{1,n} \\ \vdots & \ddots & \vdots \\ s_{n,1} & \cdots & s_{n,n} \end{pmatrix}$$

For every element of the matrix $L$ it applies that $l_{i,j} = l_{j,i}$ where $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Form the values of indexes $i$ and $j$ it can be retrieved two distinct vectors that comprises documents that are similar to $l_{i,j}$. The two retrieved vectors are $v_1[p, i]$ where $p = 1, \ldots, n$ and $v_2[q, j]$ where $q = 1, \ldots, n$ and $l_{i,j}, l_{j,i} \in L$ and $v_1[p, i], v_2[q, j] \in S$.

Our approach firstly sorts the vectors $v_i$ which are columns in matrix $S$ and presents them in the form of links or short texts. The complete algorithm is given in the form of a pseudocode depicted as below.

**Algorithm 1** The pseudocode for adaption algorithm

**Require:** $L[\cdot, \cdot], S[\cdot, \cdot]$  
**Ensure:** $v_1, v_2$

for $i = 1$ to $n$ do  
for $j = 1$ to $n$ do  
print $l_{i,j}$  
for $k = 1$ to $n$ do  
$v_1[k] \leftarrow S_{k,j}$  
$v_2[k] \leftarrow S_{j,k}$  
end for  
end for  
end for  
return $v_1, v_2$ {Return a sorted value from $S$}

The presented pseudocode in algorithm 1, returns a sorted similarity vector for each $l_{i,j}$ values form link matrix $L$. The above proposed algorithm is suitable for both personalizations where similar content related to $l_{i,j}$ is presented to a single user as well as to a group of users that share the same interest. The
latter can be achieved by clustering the similarity matrix for discovering the most interesting groups. The suitability of the approach is going to be presented in next section with the experimental results gathered by using the above mentioned algorithm.

4 Experimental Results

Experimental results are gathered from real life web site of South East European University from where we extracted approximately 117 distinct documents out of which we created the link matrix and document similarity matrix (for both Jaccard and Cosine similarity). It is worth mentioning that vector space model created with LSI is far more superior to that of tf-idf due to its ability to find hidden relationships between terms rather than performing only occurrence calculations on terms on each document. Figure 2 shows the average similarity values found by Jaccard and cosine approach on both tf-idf and LSI vector spaces.

4.1 Adaptive content recovery

the level of document recovery of the algorithm in the sense of presenting content to both an individual users and groups of users has been analyzed. In order to test the functioning of our adaptation algorithm, let us consider a use case where a user enters the home page from where he browses the information page, represented in indexer as $p_{12}$. From this point on the algorithm tries to find similar documents related to information the user seeks like: news & events, faculties, study programs etc. These documents are retrieved from similarity matrix $S$ and sorted according to their relevance, based on similarities.

\footnote{http://www.seeu.edu.mk}

Figure 2: Average Similarity for TF-IDF and LSI vector space

Figure 3: Total number of documents recovered by the adaptation algorithm in relation to average similarities for both TFIDF and LSI vector spaces.

Figure 4: The performance of the adaptation algorithm on per-user content adaptation

This is exposed in figure 3 where the total number of similar documents recovered by our adaptation algorithm with a cutoff threshold considered as their average similarities in relation to initial documents found in web site for the same average is shown.

The proposed adaptation algorithm is suitable for recovering whole pages or atomic units that are similar to $l_{ij}$ and can be presented for groups of users that share a common interest. For this purpose, clustering methods can be applied to similarity matrix $S$. The sole constraint to the algorithm here is that an additional user profiling tool is required for matching a user to a certain cluster. For creating clusters from the similarity matrix we used the VARKmeans [23] clustering algorithm.

Having all these results, the overall affectivity of the algorithm compared to the initial link state and after the adaptation algorithm applied for both per-user based adaptation and group based adaptation is illustrated as in figures 4 and 5.

From the figure it can be seen that for LSI vector
space the algorithm shows effectiveness of more than 50% and for TFIDF vector space is more than 36% compared to initial 5% in the link matrix.

The performance of the algorithm concerning adaptation for groups of users we measure the ratio of total number of documents in relation to most suitable documents in overall clusters as well as in per cluster basis. The results of the performance on overall clusters are depicted in Figure 5.

5 Conclusion and future work

In this paper, a new approach in delivering adaptive content to users or groups of users with different preferences has been introduced. The delivery of such adaptive content is achieved through the use of two auxiliary matrices that are utilized by newly presented algorithm as elaborated in section 3. The approach presented here uses a straightforward algorithm that retrieves and delivers adaptive content to visitors and omits the use of rules defined by web site administrator which in many cases can be unimportant and subjective to them. We consider, based on the above mentioned results, that the approach is suitable for general purpose adaptive web based systems. We also introduced some preliminary results concerning the documents that the algorithm retrieves, which can be of potential use to the visitor.

The future work would involve developing a web application that would implement the approach in real life as well as evaluating the degree of help that such an approach is of use to the web site visitors.

References:


